

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:)	
)	Confirmation No: 4057
Jonathan Maron)	
)	Group Art Unit: 2166
Serial No.: 10/807,060)	
)	Examiner: Johnson, Johnese T.
Filed: March 23, 2004)	
)	
For: System and Method for Providing a)	Atty. Docket No.: 100202433-2
Service in a Controlled Run-Time)	
Environment)	

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Mail Stop: Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

This Appeal Brief under 37 C.F.R. § 41.37 is submitted in support of the Notice of Appeal filed March 16, 2009, responding to the final Office Action mailed December 16, 2008.

It is not believed that extensions of time or fees are required to consider this Appeal Brief. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. §1.136(a), and any fees required therefor are hereby authorized to be charged to Deposit Account No. 08-2025.

I. Real Party in Interest

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II. Related Appeals and Interferences

There are no known related appeals or interferences that will affect or be affected by a decision in this Appeal.

III. Status of Claims

Claims 1-20 stand finally rejected. The final rejections of claims 1-20 are appealed.

IV. Status of Amendments

No claim amendments have been made subsequent to the final Office Action mailed December 16, 2008. The claims in the attached Claims Appendix reflect the present state of Applicant's claims.

V. Summary of Claimed Subject Matter

The claimed inventions are summarized below with reference numerals and references to the written description ("specification") and drawings. The subject matter described in the following appears in the original disclosure at least where indicated, and may further appear in other places within the original disclosure.

Embodiments according to independent claim 1 describe a method to provide a service in a controlled run-time environment (Figure 3, 301). Such a method comprises registering a proxy service in the controlled run-time environment (Figure 3, 301), Applicant's specification, page 6, lines 15-19, wherein the proxy service (Figure 3, 306) implements an interface defined according to the controlled run-time environment (Figure 3, 301) configured to services operating in the controlled run-time environment (Figure 3, 301) to interoperate with the proxy service (Figure 3, 306). Applicant's specification, page 6, lines 17-21. The method further comprises receiving service information by the proxy service (Figure 3, 306) from a local service (Figure 3, 305) executing in the controlled run-time environment (Figure 3, 301) via an interface method of the proxy service (Figure 3, 306) and determining by the proxy service (Figure 3, 306) whether the received service information is valid for a remote service (Figure 3, 308) requested to be invoked by the proxy service (Figure 3, 306), Applicant's specification, page 7, lines 19-26, wherein an exception is returned to the local service (Figure 3, 305) from the proxy service (Figure 3, 306) without communication of invalid service information to the remote service (Figure 3, 308) when a determination of invalid service information is made by the proxy service (Figure 3, 306). Applicant's specification, page 8, lines 21-23. The method also comprises communicating the service information to

the remote service (Figure 3, 308) from the proxy service (Figure 3, 306) when a determination of valid service information is made by the proxy service (Figure 3, 306), Applicant's specification, page 8, lines 21-23; receiving processed information from the remote service (Figure 3, 308) in response to the communicating, Applicant's specification, page 8, lines 26-28; and returning the processed information to the local service (Figure 3, 305) from the proxy service (Figure 3, 306). Applicant's specification, page 8, lines 31-32.

Embodiments according to independent claim 10 describe a system to provide a modular software service, where the system has a processor (Figure 5, 501). Such a system comprises controlled run-time environment means (Figure 3, 301; Figure 5, 501) for managing processes and service registry means (Figure 3, 302; Figure 5, 501) for registering services operating in the controlled run-time environment means (Figure 3, 301; Figure 5, 501), Applicant's specification, page 6, lines 15-19, wherein at least one registered service is a proxy service means (Figure 3, 306; Figure 5, 501). The proxy service means (Figure 3, 302; Figure 5, 501) implements an interface defined according to the controlled run-time environment means (Figure 3, 301; Figure 5, 501) for enabling services operating in the controlled run-time environment means (Figure 3, 301; Figure 5, 501) to interoperate with the proxy service means (Figure 3, 302; Figure 5, 501). Applicant's specification, page 6, lines 17-21. The proxy service means (Figure 3, 302; Figure 5, 501) comprises means for receiving service information by the proxy service means (Figure 3, 302; Figure 5, 501) from a local service (Figure 3, 305) executing in the controlled run-time environment means (Figure 3, 301; Figure 5, 501), Applicant's specification, page 7, lines 19-26; means for determining by the proxy service means

(Figure 3, 302; Figure 5, 501) whether the received service information is valid for a remote service (Figure 3, 308) requested to be invoked by the proxy service means (Figure 3, 302; Figure 5, 501), Applicant's specification, page 7, lines 19-26, wherein an exception is returned to the local service (Figure 3, 305) from the proxy service means (Figure 3, 302; Figure 5, 501) without communication of invalid service information to the remote service (Figure 3, 308) when a determination of invalid service information is made by the proxy service means (Figure 3, 302; Figure 5, 501), Applicant's specification, page 8, lines 21-23; means for communicating the service information to the remote service (Figure 3, 308) from the proxy service means (Figure 3, 302; Figure 5, 501) when a determination of valid service information is made by the proxy service means (Figure 3, 302; Figure 5, 501), Applicant's specification, page 8, lines 21-23; means for receiving processed information from the remote service (Figure 3, 308) in response to the communicated service information, Applicant's specification, page 7, lines 19-26; and means for returning the processed information to the local service (Figure 3, 305). Applicant's specification, page 8, lines 31-32.

Embodiments according to independent claim 17 describe a computer-readable storage medium that comprises executable instructions for providing a service in a controlled run-time environment. Applicant's specification, page 9, lines 3-8. The executable instructions comprises code, Applicant's specification, page 9, lines 1-4, for registering a proxy service (Figure 3, 306) in the controlled run-time environment (Figure 3, 301), Applicant's specification, page 6, lines 15-19, wherein the proxy service (Figure 3, 306) implements an interface defined according to the controlled run-time environment (Figure 3, 301) to enable services operating in the controlled run-time

environment (Figure 3, 301) to interoperate with the service, Applicant's specification, page 6, lines 17-21; code for receiving service information by the proxy service (Figure 3, 306) from a local service (Figure 3, 305) executing in the controlled run-time environment (Figure 3, 301) via a method of the proxy service (Figure 3, 306), Applicant's specification, page 7, lines 19-26; and code for determining by the proxy service (Figure 3, 306) whether the received service information is valid for a remote service (Figure 3, 308) requested to be invoked by the proxy service (Figure 3, 306), Applicant's specification, page 7, lines 19-26, wherein an exception is returned to the local service (Figure 3, 305) from the proxy service (Figure 3, 306) without communication of invalid service information to the remote service (Figure 3, 308) when a determination of invalid service information is made by the proxy service (Figure 3, 306). Applicant's specification, page 8, lines 21-23. Executable instructions further comprise code for communicating the service information to the remote service from the proxy service (Figure 3, 306) when a determination of valid service information is made by the proxy service (Figure 3, 306), Applicant's specification, page 8, lines 21-23; code for receiving processed information from the remote service in response to the communicating, Applicant's specification, page 8, lines 26-28; and code for returning the processed information to the local service (Figure 3, 305) from the proxy service (Figure 3, 306). Applicant's specification, page 8, lines 31-32.

VI. Grounds of Rejection to be Reviewed on Appeal

The following grounds of rejections are to be reviewed on appeal:

Claims 1-20 have been rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Robertson* (U.S. Patent Publication No. 2002/0188538) in view of *Pace* (U.S. Patent Publication No. 2003/0050932).

VII. Arguments

Claims 1-20 have been rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Robertson* in view of *Pace*. Appellant respectfully traverses the rejections for at least the following reasons.

A. Applicant's Claims 1-9

As provided in independent claim 1, Applicant claims:

A method to provide a service in a controlled run-time environment, comprising:

registering a proxy service in said controlled run-time environment wherein said proxy service implements an interface defined according to said controlled run-time environment configured to services operating in said controlled run-time environment to interoperate with said proxy service;

receiving service information by said proxy service from a local service executing in said controlled run-time environment via an interface method of said proxy service;

determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service;

communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service;

receiving processed information from said remote service in response to said communicating; and

returning said processed information to said local service from said proxy service.

(Emphasis added).

Appellant respectfully submits that independent claim 1 is allowable for at least the reason that *Robertson* in view of *Pace* does not disclose, teach, or suggest at least “determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service [and] communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service,” as emphasized above.

For example, *Robertson* describes a proxy or a service object to a web service. “The proxy is an object (or objects) that allows clients to access a service. The proxy is streamed to registrar 914A where it is stored in serialized form. When a consumer of service 910A looks up that service, the proxy is, in turn, streamed out to the consumer.” Paragraph 0190. “In its simplest form, a proxy merely forwards requests from a consumer, for instance, on to services 910A. However, the proxy might also contain any code that could be executed in a client environment before, after or instead of forwarding the request.” Paragraph 0193. *Robertson* (and *Pace*) fail to disclose, however, “determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without

communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service [and] communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service,” as recited in claim 1. In the Final Office Action, the Examiner refers to Fig. 15B in *Robertson* which describes part of a process to fix a reference to a target service by using smart proxies “which handle connection-related errors and re-find their respective target services” after a reference to the target service has gone stale. See para. 0283 of *Robertson* and page 11 of Final Office Action. As such, *Robertson* does not disclose determining by a proxy service whether received service information from a local service is valid for a remote service, where an exception is returned to the local service from the proxy service when a determination of invalid service information is made.

Further, *Pace* fails to remedy the deficiencies of *Robertson*. In particular, *Pace* discloses the use of a stub object which acts a proxy of an actual object; that a CDS/ADS can proxy a request for a server being requested from a target environment; and that the server would either be on the CDS/ADS or in the source environment. See paras. 0902-0903. Accordingly, *Pace* discloses that a service in one environment (e.g., target environment) makes use of a proxy service in a different environment (e.g., on the CDS/ADS). For at least this reason, *Pace* individually or in combination with *Robertson* fails to teach or suggest “determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without communication of invalid service information to said remote

service when a determination of invalid service information is made by said proxy service [and] communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service,” as recited in claim 1.

In the Final Office Action, in support of the Examiner’s rejection, the Examiner notes that *Pace* states “In an alternative embodiment, a check is made to determine if an asset adapter will handle this fault and a proxy request or redirection is performed to access the source asset.” Page 12. In response, Appellant points out that the fault being referenced is an “object fault.” “An object fault occurs when a request is made on an object that is a stub of the actual object. The stub acts as a proxy for the object, sending the request to the actual object. The actual object would typically reside in either a source environment or on an intermediate target.” See para. 0905. Accordingly, *Pace* is not directed to determining by a proxy service whether received service information is valid for a remote service requested to be invoked by the proxy service and “communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service,” as recited in claim 1.

As a result, claim 1 is patentable over *Robertson* in view of *Pace*, the rejection is improper, and the rejection of claim 1 should be overturned. Since claims 2-9 depend from claim 1 and recite additional features, claims 2-9 are allowable as a matter of law over the cited art of record.

B. Applicant's Claims 10-16

As provided in independent claim 10, Applicant claims:

A system to provide a modular software service, the system having a processor and comprising:

controlled run-time environment means for managing processes;

service registry means for registering services operating in said controlled run-time environment means, wherein at least one registered service is a proxy service means;

said proxy service means implementing an interface defined according to said controlled run-time environment means for enabling services operating in said controlled run-time environment means to interoperate with said proxy service means, said proxy service means comprising:

means for receiving service information by said proxy service means from a local service executing in said controlled run-time environment means;

means for determining by said proxy service means whether the received service information is valid for a remote service requested to be invoked by said proxy service means, wherein an exception is returned to said local service from said proxy service means without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service means;

means for communicating said service information to said remote service from said proxy service means when a determination of valid service information is made by said proxy service means;

means for receiving processed information from said remote service in response to said communicated service information; and

means for returning said processed information to said local service.

(Emphasis added).

Appellant respectfully submits that independent claim 10 is allowable for at least the reason that *Robertson* in view of *Pace* does not disclose, teach, or suggest at least “means for determining by said proxy service means whether the received service information is valid for a remote service requested to be invoked by said proxy service means, wherein an exception is returned to said local service from said proxy service

means without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service means [and] means for communicating said service information to said remote service from said proxy service means when a determination of valid service information is made by said proxy service means,” as emphasized above.

For example, *Robertson* describes a proxy or a service object to a web service. “The proxy is an object (or objects) that allows clients to access a service. The proxy is streamed to registrar 914A where it is stored in serialized form. When a consumer of service 910A looks up that service, the proxy is, in turn, streamed out to the consumer.” Paragraph 0190. “In its simplest form, a proxy merely forwards requests from a consumer, for instance, on to services 910A. However, the proxy might also contain any code that could be executed in a client environment before, after or instead of forwarding the request.” Paragraph 0193. *Robertson* (and *Pace*) fail to disclose, however, “means for determining by said proxy service means whether the received service information is valid for a remote service requested to be invoked by said proxy service means, wherein an exception is returned to said local service from said proxy service means without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service means [and] means for communicating said service information to said remote service from said proxy service means when a determination of valid service information is made by said proxy service means,” as recited in claim 10. In the Final Office Action, the Examiner refers to Fig. 15B in *Robertson* which describes part of a process to fix a reference to a target service by using smart proxies “which handle connection-related

errors and re-find their respective target services” after a reference to the target service has gone stale. See para. 0283 of *Robertson* and page 11 of Final Office Action. As such, *Robertson* does not disclose determining by a proxy service whether received service information from a local service is valid for a remote service, where an exception is returned to the local service from the proxy service when a determination of invalid service information is made.

Further, *Pace* fails to remedy the deficiencies of *Robertson*. In particular, *Pace* discloses the use of a stub object which acts a proxy of an actual object; that a CDS/ADS can proxy a request for a server being requested from a target environment; and that the server would either be on the CDS/ADS or in the source environment. See paras. 0902-0903. Accordingly, *Pace* discloses that a service in one environment (e.g., target environment) makes use of a proxy service in a different environment (e.g., on the CDS/ADS). For at least this reason, *Pace* individually or in combination with *Robertson* fails to teach or suggest “means for determining by said proxy service means whether the received service information is valid for a remote service requested to be invoked by said proxy service means, wherein an exception is returned to said local service from said proxy service means without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service means [and] means for communicating said service information to said remote service from said proxy service means when a determination of valid service information is made by said proxy service means,” as recited in claim 10.

In the Final Office Action, in support of the Examiner's rejection, the Examiner notes that *Pace* states "In an alternative embodiment, a check is made to determine if an asset adapter will handle this fault and a proxy request or redirection is performed to access the source asset." Page 12. In response, Appellant points out that the fault being referenced is an "object fault." "An object fault occurs when a request is made on an object that is a stub of the actual object. The stub acts as a proxy for the object, sending the request to the actual object. The actual object would typically reside in either a source environment or on an intermediate target." See para. 0905. Accordingly, *Pace* is not directed to determining by a proxy service whether received service information is valid for a remote service requested to be invoked by the proxy service and "means for communicating said service information to said remote service from said proxy service means when a determination of valid service information is made by said proxy service means," as recited in claim 10.

As a result, claim 10 is patentable over *Robertson* in view of *Pace*, the rejection is improper, and the rejection of claim 10 should be overturned. Since claims 11-16 depend from claim 10 and recite additional features, claims 11-16 are allowable as a matter of law over the cited art of record.

C. Applicant's Claims 17-20

As provided in independent claim 17, Applicant claims:

A computer-readable storage medium that comprises executable instructions for providing a service in a controlled run-time environment, said executable instructions comprising:

code for registering a proxy service in said controlled run-time environment wherein said proxy service implements an interface defined according to said controlled run-time environment to enable services

operating in said controlled run-time environment to interoperate with said service;

code for receiving service information by said proxy service from a local service executing in said controlled run-time environment via a method of said proxy service;

code for determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service;

code for communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service;

code for receiving processed information from said remote service in response to said communicating; and

code for returning said processed information to said local service from said proxy service.

(Emphasis added).

Appellant respectfully submits that independent claim 17 is allowable for at least the reason that *Robertson* in view of *Pace* does not disclose, teach, or suggest at least “code for determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service [and] code for communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service,” as emphasized above.

For example, *Robertson* describes a proxy or a service object to a web service. “The proxy is an object (or objects) that allows clients to access a service. The proxy is streamed to registrar 914A where it is stored in serialized form. When a consumer of

service 910A looks up that service, the proxy is, in turn, streamed out to the consumer.” Paragraph 0190. “In its simplest form, a proxy merely forwards requests from a consumer, for instance, on to services 910A. However, the proxy might also contain any code that could be executed in a client environment before, after or instead of forwarding the request.” Paragraph 0193. *Robertson* (and *Pace*) fail to disclose, however, “code for determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service [and] code for communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service,” as recited in claim 17. In the Final Office Action, the Examiner refers to Fig. 15B in *Robertson* which describes part of a process to fix a reference to a target service by using smart proxies “which handle connection-related errors and re-find their respective target services” after a reference to the target service has gone stale. See para. 0283 of *Robertson* and page 11 of Final Office Action. As such, *Robertson* does not disclose determining by a proxy service whether received service information from a local service is valid for a remote service, where an exception is returned to the local service from the proxy service when a determination of invalid service information is made.

Further, *Pace* fails to remedy the deficiencies of *Robertson*. In particular, *Pace* discloses the use of a stub object which acts a proxy of an actual object; that a CDS/ADS can proxy a request for a server being requested from a target environment;

and that the server would either be on the CDS/ADS or in the source environment. See paras. 0902-0903. Accordingly, *Pace* discloses that a service in one environment (e.g., target environment) makes use of a proxy service in a different environment (e.g., on the CDS/ADS). For at least this reason, *Pace* individually or in combination with *Robertson* fails to teach or suggest “code for determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service [and] code for communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service,” as recited in claim 17.

In the Final Office Action, in support of the Examiner’s rejection, the Examiner notes that *Pace* states “In an alternative embodiment, a check is made to determine if an asset adapter will handle this fault and a proxy request or redirection is performed to access the source asset.” Page 12. In response, Appellant points out that the fault being referenced is an “object fault.” “An object fault occurs when a request is made on an object that is a stub of the actual object. The stub acts as a proxy for the object, sending the request to the actual object. The actual object would typically reside in either a source environment or on an intermediate target.” See para. 0905. Accordingly, *Pace* is not directed to determining by a proxy service whether received service information is valid for a remote service requested to be invoked by the proxy service and “code for communicating said service information to said remote service


from said proxy service when a determination of valid service information is made by said proxy service,” as recited in claim 17.

As a result, claim 17 is patentable over *Robertson* in view of *Pace*, the rejection is improper, and the rejection of claim 17 should be overturned. Since claims 18-20 depend from claim 17 and recite additional features, claims 18-20 are allowable as a matter of law over the cited art of record.

III. Conclusion

In summary, it is Appellant’s position that Applicant’s claims are patentable over the applied cited art references and that the rejection of these claims should be overturned. Appellant therefore respectfully requests that the Board of Appeals overturn the Examiner’s rejection and allow Applicant’s pending claims.

Respectfully submitted,

By: 
Charles W. Griggers
Registration No. 47,283

Claims Appendix under 37 C.F.R. § 41.37(c)(1)(viii)

The following are the claims that are involved in this Appeal.

1. A method to provide a service in a controlled run-time environment, comprising:

registering a proxy service in said controlled run-time environment wherein said proxy service implements an interface defined according to said controlled run-time environment configured to services operating in said controlled run-time environment to interoperate with said proxy service;

receiving service information by said proxy service from a local service executing in said controlled run-time environment via an interface method of said proxy service;

determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service;

communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service;

receiving processed information from said remote service in response to said communicating; and

returning said processed information to said local service from said proxy service.

2. The method of claim 1 further comprising:
instantiating, by said controlled run-time environment, an object of a class that defines said proxy service.
3. The method of claim 2 further comprising:
instantiating, by said controlled run-time environment, said object in a partition;
and
enabling, by said controlled run-time environment, only services operating in said partition to access said proxy service.
4. The method of claim 1 wherein said communicating service information comprises:
encapsulating said service information in an extensible mark-up language (XML) file.
5. The method of claim 1 further comprising:
exposing, by said controlled run-time environment, said proxy service only when access is permitted according to security parameters.
6. The method of claim 5 wherein said exposing comprises:
determining user-level authorization from said security parameters.

7. The method of claim 5 wherein said exposing comprises:
determining process-level authorization from said security parameters.
8. The method of claim 1 further comprising:
creating a log of access to said remote service.
9. The method of claim 1 wherein said communicating said service information comprises:
performing a remote procedure call.
10. A system to provide a modular software service, the system having a processor and comprising:
controlled run-time environment means for managing processes;
service registry means for registering services operating in said controlled run-time environment means, wherein at least one registered service is a proxy service means;
said proxy service means implementing an interface defined according to said controlled run-time environment means for enabling services operating in said controlled run-time environment means to interoperate with said proxy service means, said proxy service means comprising:
means for receiving service information by said proxy service means from a local service executing in said controlled run-time environment means;
means for determining by said proxy service means whether the received service information is valid for a remote service requested to be invoked by said proxy service means, wherein an exception is returned to said local service from said proxy service

means without communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service means;

means for communicating said service information to said remote service from said proxy service means when a determination of valid service information is made by said proxy service means;

means for receiving processed information from said remote service in response to said communicated service information; and

means for returning said processed information to said local service.

11. The system of claim 10 wherein said proxy service means further comprises:

means for verifying said service information that is operable before said means for communicating.

12. The system of claim 10 wherein said proxy service means further comprises:

means for communicating with a distributed service registry to identify said remote service.

13. The system of claim 10 wherein said controlled run-time environment means instantiates said object in a partition and only permits services operating in said partition to access said proxy service.

14. The system of claim 10 where said controlled run-time environment means comprises:

security management means for exposing said proxy service only when said security management means determines access is permitted according to security parameters.

15. The system of claim 10 wherein said controlled run-time environment means comprises:

logging means for creating a log of access to said remote service.

16. The system of claim 10 wherein said means for communicating performs a remote procedure call.

17. A computer-readable storage medium that comprises executable instructions for providing a service in a controlled run-time environment, said executable instructions comprising:

code for registering a proxy service in said controlled run-time environment wherein said proxy service implements an interface defined according to said controlled run-time environment to enable services operating in said controlled run-time environment to interoperate with said service;

code for receiving service information by said proxy service from a local service executing in said controlled run-time environment via a method of said proxy service;

code for determining by said proxy service whether the received service information is valid for a remote service requested to be invoked by said proxy service, wherein an exception is returned to said local service from said proxy service without

communication of invalid service information to said remote service when a determination of invalid service information is made by said proxy service;

code for communicating said service information to said remote service from said proxy service when a determination of valid service information is made by said proxy service;

code for receiving processed information from said remote service in response to said communicating; and

code for returning said processed information to said local service from said proxy service.

18. The computer-readable storage medium of claim 17 wherein said proxy service is an object of a class that is instantiated by said controlled run-time environment.

19. The computer-readable storage medium of claim 17 wherein said executable instructions further comprise:

code for verifying said service information before said code for communicating is operable.

20. The computer-readable storage medium of claim 17 wherein said executable instructions further comprise:

code for communicating with a distributed service registry to identify said remote service.

Evidence Appendix under 37 C.F.R. § 41.37(c)(1)(ix)

There is no extrinsic evidence to be considered in this Appeal. Therefore, no evidence is presented in this Appendix.

Related Proceedings Appendix under 37 C.F.R. § 41.37(c)(1)(x)

There are no related proceedings to be considered in this Appeal. Therefore, no such proceedings are identified in this Appendix.